Computer-Aided VLSI System Design
Lab1: 4-Bit ALU with 2 Instructions

TA: 朱怡蓁 r10943012@ntu.edu.tw

Introduction

The Arithmetic logic unit (ALU) is one of the components of a computer processor. The ALU has math, logic, and some designed operations in the computer. In this lab, you will learn:

1. A synthesizable Verilog HDL code of ALU and the corresponding test bench.
2. How to run VCS simulator?

Data Preparation

1. Upload your files (Lab1.tar) to your work directory.
2. Decompress Lab1.tar with following command:

```
tar -xvf Lab1.tar
```

3. Lab1 files are shown as below:

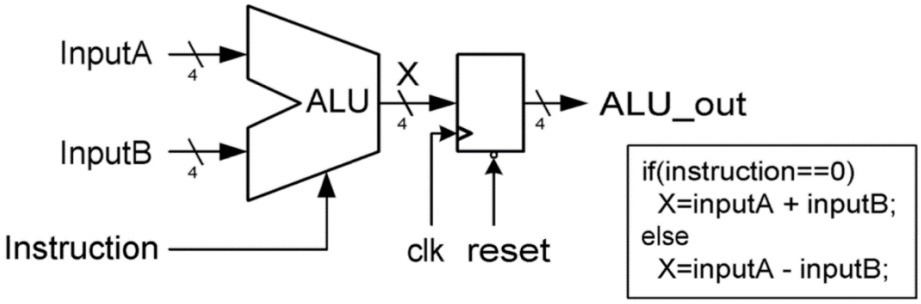
Files/Folder	Description
Lab1_alu.v	RTL code for this ALU
Lab1_test_alu.v	Test bench for this ALU
Lab1_alu_run.f	Command-line file
Lab1_test_alu_file.v	Test bench with file I/O
program.txt	Input stimulus
program_out.txt	Correct answers

Environment Setup

1. Source the default **cshrc** file:

```
source /usr/cad/synopsys/CIC/vcs.cshrc
```

Module Description



This RTL describes a simple ALU with two 4-bit input signals and 2instructions. One instruction is summation, and the other one is subtraction.

1. Type this command to enter Lab1 directory:

```
cd 112_1
```

2. View the ALU module by the following command. (You can also modify the file by this editor)

```
gedit Lab1_alu.v &
```

3. Your module looks like this

```
module ALU(alu_out,instruction,inputA,inputB,clk,reset);
output [3:0] alu_out;
input [3:0] inputA,inputB;
input instruction;
input clk,reset;

…
```

Correct Syntax Error by VCS

1. Type this command:

```
vcs -full64 Lab1_alu.v +v2k -R
```

VCS will report one error because the signal X is not declared. Please help to add the declaration for the file Lab1_alu.v .

2. Correct the code by declaration of the signal X:

- Declare signal X:

```
reg [3:0]   alu_out;
reg [3:0]   X;          // add this line
```

3. Redo step. 1, and no error occurs now.

Run Simulation with a test bench by VCS

In this lab, we use file I/O to run the simulation, you can find out how test bench works with file I/O by open Lab1_test_alu_file.v and view the following two lines:

```
$ readmemb("program.txt", program);          // Input stimulus
$ readmemb("program_out.txt", answer);       // Correct answers
```

1. Type this command

```
vcs -full64 Lab1_test_alu_file.v Lab1_alu.v +v2k -R
```

2. 224 errors are found after simulation. Please read the file alu_out.txt to find out what happens. Please modify the file Lab1_alu.v to fix these errors.
   - Correct the computing error:

```
If (!instruction)
    X=inputA+inputB;
else
    X=inputA-inputB;
```

3. Run the command in step 1 again. The simulation result is correct now.
4. Instead of the command in step 1, you can also use the command-line file to run simulation by typing this command

```
vcs -full64 -f Lab1_alu_run.f +v2k -R
```

Checkpoints

Please check with TAs before leaving this lab to make sure the following goals are accomplished and to get credits.

1. Please fix the declaration error.
2. Please fix the simulation errors.
3. Show your console:

```
[r10012@cad27 112_1]$ vcs -full64 -f Lab1_alu_run.f +v2k -R
*** Using c compiler gcc instead of cc ...
                            Chronologic VCS (TM)
         Version T-2022.06_Full64 -- Mon Sep 11 11:07:11 2023

                    Copyright (c) 1991 - 2022 Synopsys, Inc.
   This software and the associated documentation are proprietary to Synopsys,
 Inc. This software may only be used in accordance with the terms and condition
s
 of a written license agreement with Synopsys, Inc. All other use, reproduction
,
   or distribution of this software is strictly prohibited.  Licensed Products
     communicate with Synopsys servers for the purpose of providing software
    updates, detecting software piracy and verifying that customers are using
    Licensed Products in conformity with the applicable License Key for such
  Licensed Products. Synopsys will use information gathered in connection with
    this process to deliver software updates and pursue software pirates and
                                    infringers.

 Inclusivity & Diversity - Visit SolvNetPlus to read the "Synopsys Statement on
            Inclusivity and Diversity" (Refer to article 000036315 at
                        https://solvnetplus.synopsys.com)

The design hasn't changed and need not be recompiled.
If you really want to, delete file simv.daidir/.vcs.timestamp and
run VCS again.

Chronologic VCS simulator copyright 1991-2022
Contains Synopsys proprietary information.
Compiler version T-2022.06_Full64; Runtime version T-2022.06_Full64;  Sep 11 11
:07 2023

Congratulations!! Your Verilog Code is correct!!

$finish called from file "Lab1_test_alu_file.v", line 72.
$finish at simulation time                102600
           V C S   S i m u l a t i o n   R e p o r t
Time: 10260000 ps
CPU Time:      0.400 seconds;       Data structure size:   0.0Mb
Mon Sep 11 11:07:13 2023
CPU time: .460 seconds in simulation
[r10012@cad27 112_1]$
```

Submission

1. **Due Tuesday, Sep. 19, 19:00**

2. Take the snapshot of the result shown in the previous section, record it into a PDF file and submit it to NTU cool
   Title: CVSD_Lab1_studentID (E.g. CVSD_Lab1_R10943012.pdf)